Cost-sensitive Attribute Value Acquisition for Support Vector Machines*

Yee Fan Tan and **Min-Yen Kan**
School of Computing
National University of Singapore
13 Computing Drive, Singapore 117417
{tanyeefa, kanmy}@comp.nus.edu.sg

March 2010

Abstract

We consider *cost-sensitive attribute value acquisition* in classification problems, where missing attribute values in test instances can be acquired at some cost. We examine this problem in the context of the support vector machine, employing a generic, iterative framework that aims to minimize both acquisition and misclassification costs. Under this framework, we propose an attribute value acquisition algorithm that is driven by the expected cost savings of acquisitions, and for this we propose a method for estimating the misclassification costs of a test instance before and after acquiring one or more missing attribute values. In contrast to previous solutions, we show that our proposed solutions generalize to support vector machines that use arbitrary kernels. We conclude with a set of experiments that show the effectiveness of our proposed algorithm.

1 Introduction

Supervised classification involves a *training* phase, where labeled instances are collected and used to build a classifier; and a *testing* or *classification* phase, where the classifier determines the labels of new and unseen instances. In many real life problems, it is possible to control the quality of the training data but not that of the testing data. In particular, test instances often contain missing attribute values that hinders the classification task, but missing values can be acquired at some cost. As an example, consider the medical diagnosis of a patient. The doctor can order various tests such as blood tests and chest X-rays to acquire more information about the patient to make a more informed diagnosis, but each test comes at some monetary cost. In such *cost-sensitive attribute value acquisition* or *cost-sensitive acquisition* problems, the objective is to acquire attribute values in such a way to minimize both attribute acquisition and misclassification costs of the test instances.

Here, we propose a new, iterative framework for cost-sensitive attribute value acquisition for the support vector machine based classification. To the best of our knowledge, there has been no work on extending support vector machine classification to handle cost-sensitive attribute value acquisition, although prior work has investigated it in other learning models such as decision trees. This is an important contribution as support vector machines show consistently good performance over a wide range of problem classes.

Our proposed framework is illustrated in Figure 1: a process that acquires attribute values in an iterative manner. Starting with a (possibly empty) set of known attribute values A, we acquire missing attribute values A' and add them iteratively to A. Once a termination condition is satisfied, no more attributes are acquired and a final classification is performed on A. To apply this framework, we need to specify three parts:

*This work was partially supported by a National Research Foundation grant "Interactive Media Search" (grant #R-252-000-325-279).

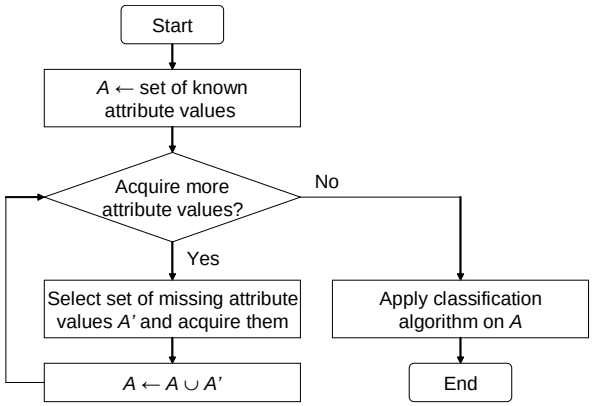

Figure 1: General iterative framework for solving classification problems with missing attribute values.

1. The classification algorithm.
2. An algorithm for selecting the set A' to acquire.
3. The termination condition.

This acquisition framework generalizes most related work in Section 2 and trivially embodies algorithms that do not consider attribute value acquisition – simply by setting the termination condition to be always be true and not looping at all. In this work, we will assume that the classification algorithm is the support vector machine, and focus on the second and third parts of the framework.

To keep the exposition easy to follow, we have purposefully taken on several assumptions that are minor and easily removed from our cost-sensitive acquisition algorithm. First, we assume that this framework is applied separately for each test instance, and that only one attribute value is acquired per iteration. Second, we apply our algorithm in the context of the standard two-class support vector machine. Finally, we also assume that the attribute acquisition and misclassification costs are known *a priori*, but note that we do not assume uniform costs for misclassifying instances of different classes. Following [Ling et al., 2006], our objective is to minimize the sum of the acquisition and misclassification costs.

Our main contribution is to provide a realization of this framework for support vector machines, something that is not seen in the published literature as far as we know. We develop an approach for estimating the misclassification cost of a test instance with missing attribute values, before and after acquiring a particular subset of these missing values. A key part of our contribution is a method that can be applied to arbitrary kernels, unlike previous work on linear classifiers that deals with missing attribute values in test instances.

We first review related work in Section 2. Section 3 is a preliminaries section that gives a brief overview on support vector machines, as well as introduce the notation we will use for the remainder of this work. To build up our cost-sensitive algorithm, we first explain in Section 4 how to compute expected misclassification costs for a test instance with missing attribute values as-is as well as the same instance with a subset of its missing attribute values acquired. Armed with these expected misclassification costs, We propose a cost-sensitive acquisition algorithm for the support vector machine classifier in Section 5 that specifies the second and third parts that are needed in the iterative acquisition framework. We evaluate our proposed algorithm in Section 6, before concluding this paper.

2 Related Work

There is a significant amount of related work dealing with classification problems involving missing values in data. [Turney, 2000] described various kinds of costs, of which the most pertinent are attribute acquisition costs and (mis)classification costs. For a problem where correct classifications incur a cost as well, [Greiner et al.,

2002] showed that it can be transformed into an equivalent problem where correct classifications incur zero cost, hence in this work we assume that no cost is incurred when a test instance is classified correctly. [Elkan, 2001] argued that *cost-sensitive classification*, where unequal misclassification costs are assigned for different classes, should be applied in situations such as when the class distribution is imbalanced. [Ling et al., 2004] extended the notion of cost-sensitive classification to include acquisition of missing attribute values in test instances, resulting in the problem we consider here. In this section, we shall focus more on algorithms handling missing values in test instances, as well as algorithms for support vector machines.

[Saar-Tsechansky and Provost, 2007] treated the underlying classifier as a black box, and proposed training a classification model on each (selected) subset of the input attributes. Test instances would then be classified using the models corresponding to its known attribute values. However, this method does not consider the possibility of acquiring missing attribute values. The active feature-value acquisition framework of [Saar-Tsechansky et al., 2009] may be seen as an extension of [Saar-Tsechansky and Provost, 2007], except that it is targeted at missing values in training data. A special case of active feature-value acquisition is applied to selective acquisition of missing attribute values in test instances [Kanani and Melville, 2008]. [Ji and Carin, 2007] formulated a partial observable Markov decision process to deal with cost-sensitive attribute value acquisition when the classifier is a hidden Markov model. For Naïve Bayes classifiers, [Chai et al., 2004] proposed classifying test instances by simply ignoring missing attribute values.

Cost-sensitive decision tree classifiers have received the most attention by researchers. [Greiner et al., 2002] proposed an algorithm for building a cost-sensitive decision tree and analyzed its theoretical properties under a probably approximately correct learning framework. [Turney, 1995] proposed using genetic algorithms, while [Zubek and Dietterich, 2002] uses Markov decision processes to generate candidate decision trees. On the other hand, [Davis et al., 2006] and [Ling et al., 2004] proposed the building of a single decision tree from the training data. [Ling et al., 2006] proposed a lazy tree formulation that improved on [Ling et al., 2004] in decision tree construction. Some of these methods generate many candidate models and have efficiency issues, while others generate only a single classification model and are efficient.

Relatively fewer works have dealt with missing values when support vector machines and other linear classifiers are considered. Training instances of a support vector machine can be weighted by their misclassification costs such that the trained support vector machine is sensitive to different misclassification costs of different classes [Osuna et al., 1997]. [Smola and Vishwanathan, 2005] and [Pelckmans et al., 2005] both builds (kernelized) support vector machines that handles missing values in training data (as opposed to test data), and does not consider the option of acquiring missing values. [Globerson and Roweis, 2006] and [Dekel and Shamir, 2008] proposed linear classifiers that are robust to missing attribute values during classification time, with [Dekel and Shamir, 2008] using a linear programming formulation which improves on [Globerson and Roweis, 2006] and is the current state-of-the-art. However, both linear classifiers 1) require the number of missing attribute values to be supplied during training, 2) does not consider acquisition of missing attribute values, and 3) cannot easily be kernelized. To the best of our knowledge, there is no support vector machine classifier that also considers the cost-sensitive acquisition of missing attribute values during classification time. Our aim is to fill in this gap by proposing such an algorithm for the support vector machine, which can be applied to any kernel satisfying Mercer's condition.

3 Preliminaries and Notation

In this section, we first give a brief overview of support vector machines and introduce our notation at the same time. For a more detailed introduction to support vector machines, we refer the reader to tutorials such as [Osuna et al., 1997] and [Burges, 1998].

3.1 Background on Support Vector Machines

Let us first ignore the issue of missing attribute values and assume all attribute values are known. Consider a classification problem where instances have n numeric attributes. A kernel function (that satisfies the Mercer's condition) is a function $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow R$ that can be expressed as $K(\mathbf{x}_i, \mathbf{x}) = \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}) \rangle$, where

$\mathbf{\Phi} : \mathbb{R}^n \to \mathcal{F}$ is a feature map from the attribute space $\mathbb{R}^n$ to some feature space $\mathcal{F}$. For a user-selected kernel function $K(\cdot, \cdot)$, a support vector machine is a linear classifier in the feature space, *i.e.*, it maps an instance $\mathbf{x} \in \mathbb{R}^n$ to a class in $\{-1, +1\}$ via $y = \text{sign}[f(\mathbf{x})]$, where the decision function $f(\cdot)$ is defined as:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{\Phi}(\mathbf{x}) \rangle + b \tag{1}$$

Training a support vector machine involves learning the weight vector $\mathbf{w} \in \mathcal{F}$ and bias $b \in \mathbb{R}$. Given training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where each $\mathbf{x}_i$ is a training instance and y_i being its class, a support vector machine can be trained by solving the following (primal) quadratic optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{N} \varepsilon_i \\ \text{Subject to} \quad & \forall i \in \{1, \ldots, N\} \quad y_i(\langle \mathbf{w}, \mathbf{\Phi}(\mathbf{x}_i) \rangle + b) \geq 1 - \varepsilon_i \\ & \forall i \in \{1, \ldots, N\} \quad \varepsilon_i \geq 0 \end{aligned} \tag{2}$$

Geometrically, a support vector machine is designed to maximize the margin $\rho = \frac{2}{\|\mathbf{w}\|_2}$. The constant C specifies the amount of misclassifications allowed for the training instances, and can either be user-specified or determined through cross-validation. Instead of solving the above primal problem directly, it is often more convenient to solve the dual optimization problem instead:

$$\begin{aligned} \text{Maximize} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle \\ \text{Subject to} \quad & \forall i \in \{1, \ldots, N\} \quad 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{3}$$

The optimal solutions of the primal and dual problems are related by:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{\Phi}(\mathbf{x}_i) \tag{4}$$

Commonly used kernels include the following:

- Linear kernel: $K(\mathbf{x}_i, \mathbf{x}) = \langle \mathbf{x}_i, \mathbf{x} \rangle$.
- Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}) = (\langle \mathbf{x}_i, \mathbf{x} \rangle + 1)^d$.
- Radial basis function (RBF) kernel: $K(\mathbf{x}'_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}'_i - \mathbf{x}'\|_2^2}{2\sigma^2}\right)$.

The linear kernel corresponds to the identity feature map $\mathbf{\Phi}(\mathbf{x}) = \mathbf{x}$. However, when the training data is non-linearly separable, a nonlinear feature map $\mathbf{\Phi}(\cdot)$ whose feature space $\mathcal{F}$ is a high or infinite dimensional space is employed. Examples of nonlinear kernel includes the polynomial kernel and the RBF kernel. In such cases, often the kernel function $K(\cdot, \cdot)$ is specified directly, leaving $\mathbf{\Phi}(\cdot)$ and $\mathcal{F}$ implicitly defined. This leaves the weight vector $\mathbf{w}$ implicitly defined as well. However, this is not a difficulty because we can train the support vector machine by solving the dual formulation (after substituting $\langle \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) \rangle$ with $K(\mathbf{x}_i, \mathbf{x}_j)$). At the same time, Equation 4 implies that we can express the decision function as:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{5}$$

Also, Equation 4 implies that:

$$\|\mathbf{w}\|_2^2 = \langle \mathbf{w}, \mathbf{w} \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

which means that the margin ρ can be computed as usual.

3.2 Posterior Probability of Classification

A trained support vector machine can be extended to provide the posterior probability that a test instance belongs to a class [Platt, 2000]. Let $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$ be the probabilities that a test instance $\mathbf{x}$ belongs to the positive and negative class respectively. Then we can fit a sigmoid function as follows:

$$p_+(\mathbf{x}) = \frac{1}{1 + \exp(Af(\mathbf{x}) + B)} \tag{7}$$

where the parameters A and B are to be learned from the training data. We can then compute $p_-(\mathbf{x}) = 1 - p_+(\mathbf{x})$.

3.3 Classifying an Instance with Missing Attribute Values

We now consider the classification of a test instance $\mathbf{x} \in \mathbb{R}^n$ with missing attribute values and without acquiring any of them. The attribute values of $\mathbf{x}$ can be partitioned into a set $\overline{M}(\mathbf{x})$ of known (or observed) attribute values and a set $M(\mathbf{x})$ of missing attribute values, such that $\{x_1, \ldots, x_n\}$ is the disjoint union of $\overline{M}(\mathbf{x})$ and $M(\mathbf{x})$. We assume that prior to classification time, a set of fixed constants $x'_1, \ldots, x'_n$ is already defined and computed. When classifying a test instance $\mathbf{x}$ with some of its attribute values missing, and without acquiring any of its missing values, we can impute (*i.e.*, replace) each missing attribute value $x_k \in M(\mathbf{x})$ with the constant x'_k before computing $f(\mathbf{x})$. Here, we assume that these fixed constants are independent of the test instance $\mathbf{x}$, but we do not impose on the exact semantics of these fixed constants. Typically, x'_k can be the mean value of the kth attribute of the training instances, or simply be the zero value. This simple approach is widely used in related work (*e.g.*, [Globerson and Roweis, 2006] and [Dekel and Shamir, 2008]) as well as support vector machine implementations (*e.g.*, the sequential minimal optimization (SMO) classifier of Weka [Witten and Frank, 2005]). Although imputing missing attribute values with fixed constants is not necessarily the best approach, imputation methods is not the focus of this work and we leave improvements in this area as future work. Thus, we simply used mean value imputation in our experiments. Similarly, for the computation of the sigmoid function, any missing attribute values of an instance are also replaced with the fixed constants before computing its posterior probability.

4 Computing Expected Misclassification Costs

Consider a test instance $\mathbf{x}$ and $A' \subseteq M(\mathbf{x})$, a subset of its missing attribute values. Let $\mathbf{x} + A'$ denote the test instance $\mathbf{x}$ with A' acquired. We define the following:

- $E[mc(\mathbf{x})]$ is the expected misclassification cost of $\mathbf{x}$.
- $E[mc(\mathbf{x} + A')]$ is the expected misclassification cost of $\mathbf{x}$ after acquiring A'.

The aim of this section is to give a way for computing the quantities $E[mc(\mathbf{x})]$ and $E[mc(\mathbf{x}+A')]$ for arbitrary $\mathbf{x}$ and A'. These quantities are two of the key quantities that drive our cost-sensitive acquisition algorithm, which we describe in the next section.

In this section, we first construct the computation of $E[mc(\mathbf{x})]$ and $E[mc(\mathbf{x} + A')]$ for an arbitrary kernel. Then, we give an interpretation for the linear kernel, and explain how these quantities can be computed efficiently in $O(n)$ time, linear in the number of attributes.

We first define $\mathbf{w}'(\mathbf{x})$, the weight vector $\mathbf{w}$ modified by the missing values of $\mathbf{x}$.

Definition 1. Let $\mathbf{x}$ be a test instance. Let $\mathbf{x}'_i$ be the training instance $\mathbf{x}_i$ with the attributes that are missing in the test instance $\mathbf{x}$ replaced by fixed constants, *i.e.*,

$$x'_{ik} = \begin{cases} x_{ik} & \text{if } x_k \in \overline{M}(\mathbf{x}) \\ x'_{ik} & \text{otherwise} \end{cases} \tag{8}$$

The *modified weight vector* $\mathbf{w}'(\mathbf{x})$ is defined such that:

$$\|\mathbf{w}'(\mathbf{x})\|_2^2 = \langle \mathbf{w}'(\mathbf{x}), \mathbf{w}'(\mathbf{x}) \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}'_i, \mathbf{x}'_j) \tag{9}$$

□

For the instance $\mathbf{x}+A'$, we define its modified weight vector $\mathbf{w}'(\mathbf{x}+A')$ by taking $\overline{M}(\mathbf{x}+A') = \overline{M}(\mathbf{x}) \cup A'$. Also, notice that Equation 9 is defined in a manner that is analogous to Equation 6.

The quantity $\|\mathbf{w}'(\mathbf{x})\|_2$ can be interpreted as the amount of information that is remaining from $\|\mathbf{w}'\|_2$ after taking into account of information that is lost due to the missing attribute values. This aspect will be made obvious when an interpretation for $\mathbf{w}'(\mathbf{x})$ for the linear kernel is made later.

Definition 2. The *uncertainty factor* for a test instance $\mathbf{x}$ is defined by:

$$u(\mathbf{x}) = \frac{\|\mathbf{w}'(\mathbf{x})\|_2}{\|\mathbf{w}\|_2} \tag{10}$$

□

Empirically, we found that $\|\mathbf{w}'(\mathbf{x})\|_2 \leq \|\mathbf{w}'\|_2$ is almost always satisfied, which means that we almost always have $0 \leq u(\mathbf{x}) \leq 1$. These inequalities are always satisfied when the linear kernel is used.

We now explain the computation for $E[mc(\mathbf{x})]$, the estimated misclassification cost for a test instance $\mathbf{x}$ with missing attribute values. Let mc_+ and mc_- be the misclassification cost for misclassifying a positive and negative instance, respectively. Suppose the support vector machine classifies $\mathbf{x}$ to be positive, *i.e.*, $f(\mathbf{x}) \geq 0$. Then according to the sigmoid function of Equation 7, the posterior probability of $\mathbf{x}$ belonging to the positive class is $p_+(\mathbf{x})$. However, the computation of $p_+(\mathbf{x})$ does not take missing attribute values into account. Hence, the certainty of the classification is reduced by the uncertainty factor $u(\mathbf{x})$, *i.e.*, meaning that the certainty of is now $u(\mathbf{x}) \cdot p_+(\mathbf{x})$. Thus the uncertainty of the classification is $1 - u(\mathbf{x}) \cdot p_+(\mathbf{x})$. Hence, we compute the expected misclassification cost as $(1-u(\mathbf{x}) \cdot p_+(\mathbf{x})) \cdot mc_-$. Similarly, if the support vector machine classifies $\mathbf{x}$ to be negative, then we compute the expected misclassification cost as $(1-u(\mathbf{x}) \cdot p_-(\mathbf{x})) \cdot mc_+$. The following definition summarizes the computation of the expected misclassification cost of an test instance $\mathbf{x}$.

Definition 3. Let mc_+ and mc_- be the misclassification cost for misclassifying a positive and negative instance, respectively. The *expected misclassification cost* of a test instance $\mathbf{x}$ is defined as:

$$E[mc(\mathbf{x})] = \begin{cases} (1 - u(\mathbf{x}) \cdot p_+(\mathbf{x})) \cdot mc_- & \text{if } f(\mathbf{x}) \geq 0 \\ (1 - u(\mathbf{x}) \cdot p_-(\mathbf{x})) \cdot mc_+ & \text{otherwise} \end{cases} \tag{11}$$

□

Again, it is straightforward to modify the above definition for $E[mc(\mathbf{x} + A')]$:

$$E[mc(\mathbf{x} + A')] = \begin{cases} (1 - u(\mathbf{x} + A') \cdot p_+(\mathbf{x})) \cdot mc_- & \text{if } f(\mathbf{x}) \geq 0 \\ (1 - u(\mathbf{x} + A') \cdot p_-(\mathbf{x})) \cdot mc_+ & \text{otherwise} \end{cases} \tag{12}$$

The inequality $\|\mathbf{w}'(\mathbf{x})\|_2 \leq \|\mathbf{w}'(\mathbf{x} + A')\|_2$ always holds for the linear kernel and should hold most of the time for nonlinear kernels. If this inequality holds, then we also have $u(\mathbf{x}) \geq u(\mathbf{x} + A')$, and hence $E[mc(\mathbf{x})] \geq E[mc(\mathbf{x} + A')]$. In other words, the acquisition of missing attribute values should lead to a

decrease in expected misclassification cost. By this token, the difference of $E[mc(\mathbf{x})]$ and $E[mc(\mathbf{x}+A')]$ can be exploited to determine how useful it is to acquire A'. This difference is the key element for constructing the cost-sensitive acquisition algorithm in the next section.

Earlier, we mentioned that the quantity $\|\mathbf{w}'(\mathbf{x})\|_2$ can be interpreted as the amount of information remaining after taking into account the information that is lost from the missing attribute values. In the following, we will make this interpretation obvious in the context of the linear kernel. As this quantity will be computed many times in the cost-sensitive acquisition algorithm, we also suggest a way to compute it efficiently when a non-linear kernel is used.

4.1 Modified Weight Vector for Linear Kernel

Recall that the linear kernel is $K(\mathbf{x}_i, \mathbf{x}) = \langle \mathbf{x}_i, \mathbf{x} \rangle$, whose feature map is $\mathbf{\Phi}(\mathbf{x}) = \mathbf{x}$. For a support vector machine that uses a linear kernel, its decision function can be simplified as follows:

$$f(\mathbf{x}) = \sum_{k=1}^{n} w_k x_k + b \tag{13}$$

If we assume that the attribute values in both training and test instances have been appropriately normalized to the same scale, then the magnitude of the weight w_k of each attribute x_k indicates the relevance or importance of that attribute. Thus, if the missing attribute values of a test instance has weights of large magnitude, then its classification would be highly uncertain.

For a test instance $\mathbf{x}$, we can define its modified weight vector explicitly as $\mathbf{w}'(\mathbf{x}) = (w'_1(\mathbf{x}), \ldots, w'_n(\mathbf{x}))$, where

$$w'_k(\mathbf{x}) = \begin{cases} w_k & \text{if } x_k \in \overline{M}(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Theorem 1. When the linear kernel is used, both the modified weight vectors $\mathbf{w}'(\mathbf{x})$ as defined by Equation 9 and Equation 14 compute the same value for $\|\mathbf{w}'(\mathbf{x})\|_2$, *i.e.*,

$$\|\mathbf{w}'(\mathbf{x})\|_2^2 = \sum_{k:x_k \in \overline{M}(\mathbf{x})} w_k^2 \tag{15}$$

Proof. When the linear kernel is used, we have that for two training instances $\mathbf{x}_i$ and $\mathbf{x}_j$, $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{k:x_k \in M(\mathbf{x})} x'_{ik} + \sum_{k:x_k \in \overline{M}(\mathbf{x})} x_{ik}$. Note that the first term on the right hand side is a constant.

For $\mathbf{w}'(\mathbf{x})$ as defined by Equation 9, we have

$$\|\mathbf{w}'(\mathbf{x})\|_2^2 = \sum_{k:x_k \in M(\mathbf{x})} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x'_{ik} x'_{jk} + \sum_{k:x_k \in \overline{M}(\mathbf{x})} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_{ik} x_{jk} \tag{16}$$

From the dual formulation of the support vector machine, $\sum_{i=1}^{N} \alpha_i y_i = 0$, and thus the first term on the right hand side of Equation 16 is zero. Also, when the linear kernel is used, we have $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$, which implies $w_k = \sum_{i=1}^{N} \alpha_i y_i x_{ik}$. Therefore, the second term on the right hand side of Equation 16 becomes $\sum_{k:x_k \in \overline{M}(\mathbf{x})} w_k^2$. Hence, Equation 15 is satisfied.

For $\mathbf{w}'(\mathbf{x})$ as defined by Equation 14, it is easy to see that Equation 15 is satisfied as well. □

The above theorem states that we can compute $\|\mathbf{w}'(\mathbf{x})\|_2$ using the form of $\mathbf{w}'(\mathbf{x})$ that is explicitly defined in Equation 14. As this explicit form of $\mathbf{w}'(\mathbf{x})$ is formed by zeroing the weights of $\mathbf{w}$ corresponding to the missing attribute values of $\mathbf{x}$, the quantity $\|\mathbf{w}'(\mathbf{x})\|_2 = \sum_{k:x_k \in \overline{M}(\mathbf{x})} w_k^2$ has a particularly intuitive interpretation. It is less than $\|\mathbf{w}\|_2 = \sum_{k=1}^{N} w_k^2$ and represents the amount of useful information remaining after taking into account the missing attribute values of $\mathbf{x}$.

Given the explicit representation of the modified weight vector $\mathbf{w}'(\mathbf{x})$ for the linear kernel, we can see that computing $\|\mathbf{w}'(\mathbf{x})\|$, and hence $E[mc(\mathbf{x})]$, incurs $O(n)$ time, linear in the number of attributes. The same can be said for the computation of $E[mc(\mathbf{x}+A')]$ as well.

A final remark we make is that this method for computing $E[mc(\mathbf{x})]$ and $E[mc(\mathbf{x}+A')]$ can be applied to any linear classifier whose decision function is of the form Equation 13. It is not necessary for the linear classifier to be a support vector machine.

4.2 Modified Weight Vector for Nonlinear Kernel

The main challenge for the computation of $\|\mathbf{w}'(\mathbf{x})\|_2$ for a nonlinear kernel is that it requires $O(M^2n)$ time to compute using Equation 6 requires $O(M^2n)$ time, where M is the number of support vectors in the support vector machine. This makes our proposed acquisition algorithm of computationally expensive during classification time as many computations of expected misclassification costs may be required. This is especially so when M is large and when the test instance $\mathbf{x}$ has many missing attribute values, which are typical characteristics for large datasets. Using Equation 6 to compute $\|\mathbf{w}'(\mathbf{x})\|_2$ will make our proposed acquisition algorithm impractical for these cases.

One approach around the problem of computational complexity is to reduce the number of support vectors in the trained support vector machine. This approach has been studied in papers such as [Burges, 1996], [Lin and Lin, 2003], [Apolloni et al., 2004], and [Nguyen and Ho, 2005]. However, we choose to use another approach that allows us to compute $\|\mathbf{w}'(\mathbf{x})\|_2$ in $O(n)$ time, linear in the number of attributes. This approach approximates the computation of the kernel function $K(\mathbf{x}'_i, \mathbf{x}'_j)$ using "weak kernels" [Fung et al., 2007]. Let $K_k(\mathbf{x}'_i, \mathbf{x}'_j)$ be the kernel function computed using only the kth attribute of $\mathbf{x}'_i$ and $\mathbf{x}'_j$. For example, if $K(\mathbf{x}'_i, \mathbf{x}'_j)$ is the radial basis function (RBF) kernel defined by $K(\mathbf{x}'_i, \mathbf{x}'_j) = \exp\left(-\frac{\|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2}{2\sigma^2}\right)$, then $K_k(\mathbf{x}'_i, \mathbf{x}'_j) = \exp\left(-\frac{\|x'_{ik} - x'_{jk}\|_2^2}{2\sigma^2}\right)$. Then we approximate the computation of $K(\mathbf{x}'_i, \mathbf{x}'_j)$ by:

$$K(\mathbf{x}'_i, \mathbf{x}'_j) \approx \sum_{k=1}^{n} \gamma_k K_k(\mathbf{x}'_i, \mathbf{x}'_j) \tag{17}$$

where the constants $\gamma_1, \ldots, \gamma_n$ are to be determined. Thus, Equation 9 becomes:

$$\|\mathbf{w}'(\mathbf{x})\|_2^2 = \langle \mathbf{w}'(\mathbf{x}), \mathbf{w}'(\mathbf{x}) \rangle \approx \sum_{k=1}^{n} \gamma_k t_k(\mathbf{x}) \tag{18}$$

where:

$$t_k(\mathbf{x}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K_k(\mathbf{x}'_i, \mathbf{x}'_j) \tag{19}$$

Note that each of the $t_k(\mathbf{x})$ terms can take on only one of two possible values, which depends on whether $x_k \in M(\mathbf{x})$, *i.e.*, whether the kth attribute in the test instance $\mathbf{x}$ is known or missing. This allows us to precompute these terms during training, by selecting a subset of missing patterns out of the 2^n possible patterns and computing their corresponding actual $\|\mathbf{w}'(\mathbf{x})\|_2^2$ values, and determining the values of $\gamma_1, \ldots, \gamma_n$ using a standard linear regression algorithm. Subsequently, during classification time, when we need to estimate $\|\mathbf{w}'(\mathbf{x})\|_2$ for a test instance $\mathbf{x}$ with missing values, we use these precomputed approximations. The same precomputed approximations can also be used to compute $\|\mathbf{w}'(\mathbf{x}+A')\|_2$ for any A'. In this way, each computation of $\|\mathbf{w}'(\mathbf{x})\|_2$ or $\|\mathbf{w}'(\mathbf{x}+A')\|_2$, and hence $E[mc(\mathbf{x})]$ or $E[mc(\mathbf{x}+A')]$, takes only $O(n)$ time, linear in the number of attributes. In this work, we always use this approximation when computing expected misclassification costs for a support vector machine using a nonlinear kernel.

5 A Cost-sensitive Attribute Value Acquisition Algorithm

Given a method for computing $E[mc(\mathbf{x})]$ and $E[mc(\mathbf{x}+A')]$ efficiently, we are now in a position to describe an algorithm that uses a trained support vector machine to perform cost-sensitive attribute value acquisition under the framework of Figure 1. This algorithm defines the attribute value selection algorithm and the termination condition, the second and third parts in the iterative framework.

Note that the expression $E[mc(\mathbf{x})] - E[mc(\mathbf{x}+A')]$ gives the expected decrease in misclassification cost, and the aim here is to minimize the total cost of acquisitions and misclassifications. Hence, we define the following.

Definition 4. Consider a test instance $\mathbf{x}$ and $A' \subseteq M(\mathbf{x})$, a subset of its missing attribute values. Let $ac(A')$ be the acquisition cost of the attribute values A', which is assumed to be given in the cost-sensitive acquisition problem. The *expected reduction in total cost* for acquiring A' is defined by:

$$E[rc(A')] = (E[mc(\mathbf{x})] - E[mc(\mathbf{x}+A')]) - ac(A') \tag{20}$$

□

Our cost-sensitive attribute value acquisition algorithm is driven by the expected reduction in total cost that can be achieved by acquiring missing attribute values in test instances. This approach follows that of [Ling et al., 2006]. Here, we consider a strategy of sequentially acquiring one attribute value in each iteration, until there is no more attribute value x_k with positive $E[rc(\{x_k\})]$. This acquisition algorithm is shown in Algorithm 1.

Algorithm 1 Cost-sensitive attribute value acquisition algorithm.

Input: A test instance $\mathbf{x}$

1: **loop**
2: Select the attribute value $x_k \in M(\mathbf{x})$ with the maximum $E[rc(\{x_k\})]$
3: **if** $E[rc(\{x_k\})] \leq 0$ **then**
4: **break**
5: $\mathbf{x} \leftarrow \mathbf{x} + \{x_k\}$ {Acquire the attribute value x_k}
6: Classify $\mathbf{x}$

We note that it is possible to employ batch acquisition strategies. However, as our focus in this work is the computation of $E[mc(\mathbf{x})]$ and $E[mc(\mathbf{x}+A')]$, thereby facilitating the construction of a cost-sensitive attribute value acquisition algorithm for a trained support vector machine, therefore we leave batch acquisition strategies as future work.

6 Evaluation

In this section, we establish empirically the effectiveness of our proposed support vector machine based benefit function. As this evaluation is difficult to perform directly, we perform this evaluation indirectly through our proposed cost-sensitive attribute value acquisition algorithm.

To establish the above goal, we compared our cost-sensitive acquisition algorithm against a baseline that randomly acquires a proportion (25%, 50%, or 75%) of the missing attribute values in each test instance. This evaluation is performed separately on support vector machines with three different kernels: linear, polynomial, and radial basis function (RBF). The classifiers are made sensitive to imbalanced misclassification costs by modifying the primal optimization problem, by replacing the $\sum_{i=1}^{N} \varepsilon_i$ part of the objective function with $\sum_{i=1}^{N} \lambda_i \varepsilon_i$, where λ_i is a weight proportional to the misclassification cost of the training instance $\mathbf{x}_i$ [Osuna et al., 1997].

We performed the experiments using ten commonly used datasets from the UCI machine learning repository [Asuncion and Newman, 2007], all of which consists of entirely numeric or binary attributes. All datasets

Dataset	Description	Attributes	Instances	Class distribution
ECOLI	Protein localization sites	8	336	143 / 193
GLASS	Glass identification	9	214	51 / 163
HEART	Cleveland heart disease	13	303	139 / 164
HEPATITIS	Hepatitis domain	34	351	126 / 225
IONOSPHERE	Ionosphere radar data	34	351	126 / 225
PIMA	Pima Indians diabetes	8	768	268 / 500
SONAR	Sonar signals, mines versus rocks	60	208	97 / 111
SPAM	Emails spam detection	20	4601	1813 / 2788
THYROID	Thyroid disease	20	3772	284 / 3488
WDBC	Wisconsin diagnostic breast cancer	30	569	212 / 357

Table 1: Summary of the datasets.

with more than two classes were converted into two-class problems by merging classes. By convention, the minority class is set to be the positive class. A summary of the datasets is given in Table 1. For each dataset, we normalized the instances such that each attribute is supported on $[-1, +1]$ with mean 0, and split each dataset into equal-sized training and testing sets with the same class distribution in each set. Following [Ling et al., 2006], we used the training set as-is, and randomly mark a proportion (20%, 40%, 60%, 80%, or 100%) of the attribute values in the testing set as missing. At classification time, all unacquired missing attribute values of a (normalized) test instance are replaced by the zero value before its class is determined. We follow the protocol of [Ling et al., 2006] and set the misclassification costs of a positive instance and a negative instance to be 600 and 200 respectively. Where available, we used the acquisition costs supplied in [Turney, 1995]; and for the remainder of the datasets, we set the acquisition cost of each attribute to be a random number between 1 and 100. The evaluation measure is the total cost of acquisitions and misclassifications.

In our experiments, we used the LIBSVM implementation for support vector machines [Chang and Lin, 2001]. When training a classifier, five-fold cross validation is used to perform parameter tuning. For each acquisition algorithm and classifier combination, we repeated the training and testing process 20 times, and record the average total cost per test instance. The results of our experiments are shown in Figure 2 for the linear kernel, Figure 3 for the polynomial kernel, and Figure 4 for the RBF kernel.

The results show that generally, in all datasets, our cost-sensitive acquisition algorithm can achieve a much lower total cost compared to randomly acquiring a proportion of attribute values, especially when a greater proportion of attribute values is missing. We found that our cost-sensitive acquisition algorithm may incur acquisition costs greater or lesser than that of random acquisition, but our acquisition algorithm generally makes suitable trade-offs between acquisitions and misclassifications, allowing total costs to be minimized. We performed one-tail Wilcoxon signed-rank tests comparing the average total costs of our cost-sensitive acquisition algorithm versus each of the three random acquisition algorithms. The statistical significance tests were performed separately for each of the three kernels (linear, polynomial, and RBF) and each of the five proportions of missing attribute values (20%, 40%, 60%, 80%, and 100%). The results of the significance tests are shown in Table 2, which indicates that the differences are statistically significant in all classifiers. For the vast majority of the forty-five cases, we record a significant difference at the $p < 0.01$ level, although four of them were significant only at the $p < 0.05$ or $p < 0.10$ level. The statistical significance is seen for all of the linear, polynomial, and RBF kernels.

In summary, our experiments have established that our acquisition algorithm is effective, when compared to random acquisition.

7 Conclusion

In this paper, we described a general iterative framework for solving the cost-sensitive attribute value acquisition problem. We proposed a realization of the framework when the classifier is the support vector machine

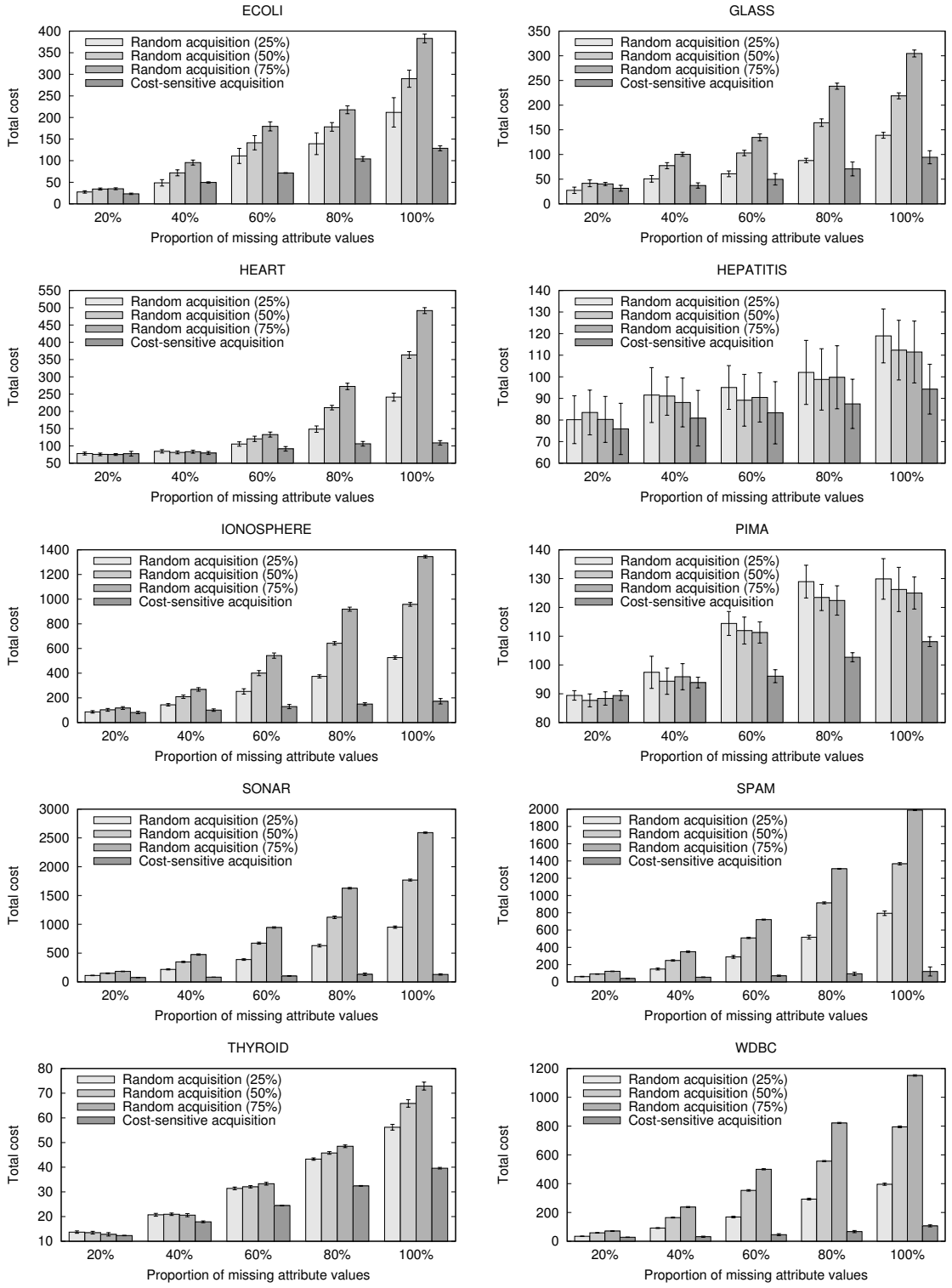

Figure 2: Average total cost per test instance for the linear kernel. Error bars indicate one standard deviation.

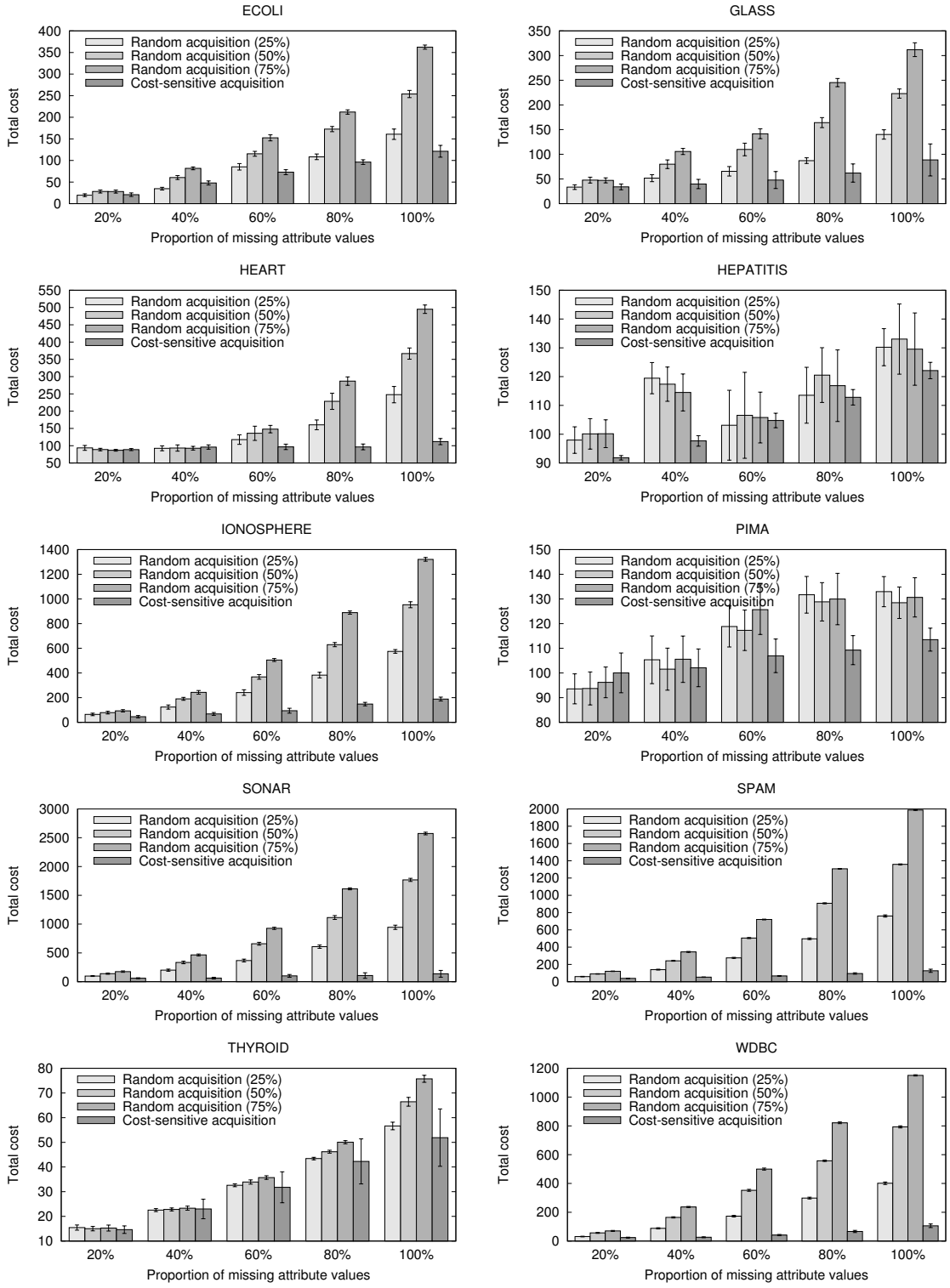

Figure 3: Average total cost per test instance for the polynomial kernel. Error bars indicate one standard deviation.

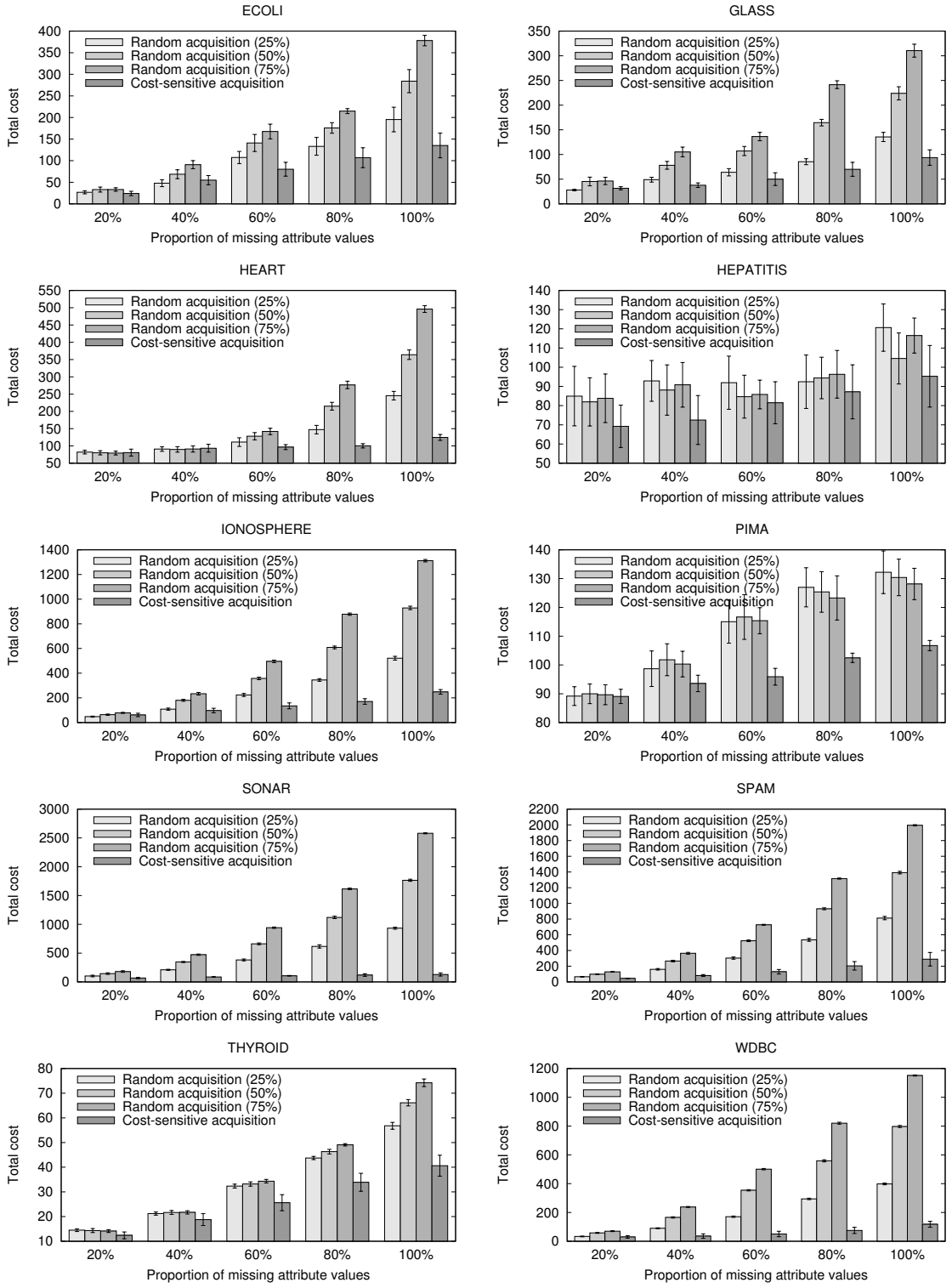

Figure 4: Average total cost per test instance for the RBF kernel. Error bars indicate one standard deviation.

Acquisition algorithm	Proportion of missing attribute values				
	20%	40%	60%	80%	100%
Linear kernel					
Random (25%)	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Random (50%)	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Random (75%)	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Polynomial kernel					
Random (25%)	$p < 0.05$	$p < 0.05$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Random (50%)	$p < 0.01$	$p < 0.05$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Random (75%)	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
RBF kernel					
Random (25%)	$p < 0.10$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Random (50%)	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$
Random (75%)	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$

Table 2: p-values of the one-tail Wilcoxon signed-rank tests between our cost-sensitive acquisition algorithm and random acquisition.

by computing the expected reduction in total costs for acquiring attribute values. A key part in our algorithm is a method for estimating the expected misclassification cost of a test instance with missing attribute values using the weight vector of the support vector machine. Another key aspect in our paper is the generalization of our algorithms such that they can be applied to support vector machines with arbitrary kernels, and not merely limited to linear support vector machines. Our experiments show that our cost-sensitive acquisition algorithm is effective on linear classifiers as well as support vector machines using the RBF kernel.

References

[Apolloni et al., 2004] Apolloni, B., Marinaro, M., and Tagliaferri, R. (2004). An algorithm for reducing the number of support vectors. In *Italian Workshop on Neural Nets (WIRN VIETRI)*, pages 99–105.

[Asuncion and Newman, 2007] Asuncion, A. and Newman, D. J. (2007). UCI machine learning repository. Available at http://archive.ics.uci.edu/ml/.

[Burges, 1996] Burges, C. J. C. (1996). Simplified support vector decision rules. In *International Conference on Machine Learning (ICML)*, pages 71–77.

[Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[Chai et al., 2004] Chai, X., Deng, L., Yang, Q., and Ling, C. X. (2004). Test-cost sensitive naive bayes classification. In *IEEE International Conference on Data Mining (ICDM)*, pages 51–58.

[Chang and Lin, 2001] Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[Davis et al., 2006] Davis, J. V., Ha, J., Rossbach, C. J., Ramadan, H. E., and Witchel, E. (2006). Cost-sensitive decision tree learning for forensic classification. In *European Conference on Machine Learning (ECML)*, pages 622–629.

[Dekel and Shamir, 2008] Dekel, O. and Shamir, O. (2008). Learning to classify with missing and corrupted features. In *International Conference on Machine Learning (ICML)*, pages 216–223.

[Elkan, 2001] Elkan, C. (2001). The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 973–978.

[Fung et al., 2007] Fung, G., Rosales, R., and Rao, R. B. (2007). Feature selection and kernel design via linear programming. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 786–791.

[Globerson and Roweis, 2006] Globerson, A. and Roweis, S. (2006). Nightmare at test time: robust learning by feature deletion. In *International Conference on Machine Learning (ICML)*, pages 353–360.

[Greiner et al., 2002] Greiner, R., Grove, A. J., and Roth, D. (2002). Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139(2):137–174.

[Ji and Carin, 2007] Ji, S. and Carin, L. (2007). Cost-sensitive feature acquisition and classification. *Pattern Recognition*, 40(5):1474–1485.

[Kanani and Melville, 2008] Kanani, P. and Melville, P. (2008). Prediction-time active feature-value acquisition for customer targeting. In *NIPS Workshop on Cost Sensitive Learning*.

[Lin and Lin, 2003] Lin, K.-M. and Lin, C.-J. (2003). A study on reduced support vector machines. *IEEE Transactions on Neural Networks*, 14(6):1449–1559.

[Ling et al., 2006] Ling, C. X., Sheng, V. S., and Yang, Q. (2006). Test strategies for cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18(8):1055–1067.

[Ling et al., 2004] Ling, C. X., Yang, Q., Wang, J., and Zhang, S. (2004). Decision trees with minimal costs. In *International Conference on Machine Learning (ICML)*, page 69.

[Nguyen and Ho, 2005] Nguyen, D. and Ho, T. (2005). An efficient method for simplifying support vector machines. In *International Conference on Machine Learning (ICML)*, pages 617–624.

[Osuna et al., 1997] Osuna, E. E., Freund, R., and Girosi, F. (1997). Support vector machines: Training and applications. Technical Report AIM-1602, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[Pelckmans et al., 2005] Pelckmans, K., De Brabanter, J., Suykens, J. A. K., and De Moor, B. (2005). Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684–692.

[Platt, 2000] Platt, J. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74.

[Saar-Tsechansky et al., 2009] Saar-Tsechansky, M., Melville, P., and Provost, F. (2009). Active feature-value acquisition. *Management Science*, 55(4):664–684.

[Saar-Tsechansky and Provost, 2007] Saar-Tsechansky, M. and Provost, F. (2007). Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1623–1657.

[Smola and Vishwanathan, 2005] Smola, A. J. and Vishwanathan, S. V. N. (2005). Kernel methods for missing variables. In *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 325–332.

[Turney, 1995] Turney, P. D. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research (JAIR)*, 2:369–409.

[Turney, 2000] Turney, P. D. (2000). Types of cost in inductive concept learning. In *ICML 2000 Workshop on Cost-Sensitive Learning*, pages 15–21.

[Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, second edition.

[Zubek and Dietterich, 2002] Zubek, V. B. and Dietterich, T. G. (2002). Pruning improves heuristic search for cost-sensitive learning. In *International Conference on Machine Learning (ICML)*, pages 19–26.